REMARKS

Claims 1-44 are now pending in this application. In the Office Action, the Examiner rejected Claims 10, 11, 25, 28 and 37 under 35 U.S.C. § 112, first paragraph, as lacking enablement. Claims 1, 3, 4, 6-11, 13-16, 19, 25-28, 30-33, 36, 37, 39-42, and 44 have been rejected under 35 U.S.C. § 112, second paragraph, as indefinite. Claims 1-8, 10-44 have been rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,252,589 to Bjorn C. Rettig et al. ("Rettig"). Claim 9 has been rejected under 35 U.S.C. § 103(a) as being unpatentable over Rettig in view of U.S. Patent No. 6,807,558 to Gregory P. Hassett et al. ("Hassett"). Lastly, the Examiner has objected to a typographical error in paragraph 13 of the specification, which has been corrected in this amendment and response.

The Applicant respectfully traverses the rejections of Claims 1-44 as set forth the Office Action. Without admitting the propriety of the rejections under 35 U.S.C. § 112, first and second paragraphs, Claims 1, 3, 4, 6-11, 13-19, 25-33, 36-42, and 44 have been amended to more particularly point out and distinctly claim the subject matter which the applicant regards as the invention. Further, Applicant respectfully submits that Rettig, Hassett, and knowledge in the art at the time of making the invention, either alone or in combination, fail to teach or suggest any technology for dynamically verifying resource module compatibility with an operating system as recited in Claims 1-44 as presently presented. Pursuant to 37 C.F.R. § 1.111, and for the reasons set forth below, the Applicant respectfully requests reconsideration and allowance of this application.

Before discussing in detail the reasons why Applicant believes that Claims 1-44 as presently presented are allowable, brief descriptions of the disclosed embodiments of the present invention and the cited and applied references are provided in the following discussion. The Applicant submits that the following discussion of the disclosed embodiments of Applicant's

MSFT\16734AM1.DOC

invention and the discussion of the differences among the disclosed embodiments and the teachings in the applied references are not provided to define the scope or interpretation of any of the claims. Instead, such discussed differences are provided to help the United States Patent and Trademark Office (hereinafter "the Office") better appreciate important claim distinctions discussed thereafter.

Summary:

The present invention is an improvement over conventional Multiple Language User Interface ("MUI") operating systems in that it facilitates the use of MUI functionality even after upgrading the MUI operating system. As the Applicant explained in the specification of this application, in conventional MUI-based operating systems MUI functionality is often reduced after an upgrade to a new version because the operating system returns to the default language. The alternate resources corresponding to the multiple languages may have been localized from other versions of the operating system, and they may or may not be compatible with the current operating system. Thus, it is safer not to load them in the current operating system as they may cause the system to hang or otherwise become inoperable.

Suppose, for example, that the default language of a conventional MUI-based operating system is English, and the alternate language is Chinese. The operating system has been recently upgraded to version 3.1. If the resource loader receives a request to load a resource from an alternate (Chinese) resource module that was localized from a version 3.0 default (English) resource module, the resource loader will not load the Chinese language resource from the alternate resource module, but will instead use the English language resource from the operating system's current default resource module, i.e., the recently upgraded version 3.1 default resource module.

This scenario occurs because the conventional MUI-based operating system assumes, often incorrectly, that the alternate resource module is no longer compatible with the upgraded version of the operating system, i.e., that the Chinese language resource is out-of-date with the upgraded default English language resources of the operating system. The assumption is often

-12-

incorrect because, in many operating system upgrades, some or most of the specific interfaces that the operating system provides remain unchanged. For example, the File menu for version 3.1 of an operating system may be unchanged from the 3.0 version. Accordingly, many of the alternate resource modules corresponding to those specific interfaces remain unchanged as well, and thus the alternate resources can be safely loaded despite the upgrade.

To address this problem, the present invention provides a method for dynamically verifying the compatibility of an alternate resource with an operating system. In accordance with the method, a resource loader obtains a request to load a resource from an alternate resource module, where the alternate resource module corresponds to a specific interface translated into one of the multiple languages supported by the system. The resource loader obtains the resource content of the default resource module from which the alternate resource module was localized. The default resource module generally includes default resource content code and a variety of resources, including data structures, templates, definition procedures, management routines, icon maps, and so forth, associated with a particular resource, such as a menu, window, or dialog box, and is typically in English.

Depending upon when localization of the alternate resource module occurred, the default resource module from which the alternate resource module was localized may or may not be at the same upgrade level as the operating system's current default resource module. In a typical embodiment, the default resource module includes an identifier including the version information of the operating system version with which the default resource module is compatible. Although unlikely, the default resource module may be at a higher upgrade level than the operating system's current default resource module. The more typical scenario is when the default resource module is at the same or lower upgrade level than the operating system's current default resource module. If they are at the same upgrade level, then of course they are compatible, and any alternate resource modules localized from that default resource module may be safely loaded. If they are not at the same upgrade level, however, then the alternate resource module may or may not be compatible with the operating system.

In order to dynamically verify whether they are compatible, the resource loader also obtains the resource content of the current operating system's default resource module. The resource loader compares the resource content of the default resource module from which the alternate resource module was localized with the resource content of the current operating system's default resource module, i.e. the current default resource module. Since both the default and current default resource modules contain resources written in the same language, typically in English, the resource content will be the same if the specific interface, i.e., the menus, windows, dialog boxes, is unchanged. If they are the same, then any of the alternate resource modules localized from the default resource module, such as those in Chinese, French, etc., are still compatible with the operating system and may be safely loaded. Otherwise, the alternate resource modules localized from the default resource module may be incompatible, most likely due to an operating system upgrade that changed the specific interface, and may not be safely loaded.

As an example, suppose that a specific interface representing the menu options of the File menu has changed between version 3.0 of an operating system and 3.1 of an operating system, but that another specific interface representing the menu options of the Edit menu has not. The resource content of the default and current default resource modules corresponding to the File menu will be different, whereas the resource content of the default and current default resource modules corresponding to the Edit menu will be the same. Accordingly, the alternate resource modules corresponding to the File menu are incompatible and cannot be safely loaded, whereas the alternate resource modules corresponding to the Edit menu are compatible and safe to load.

To facilitate the comparison of the resource content of the default and current default resource modules, in one embodiment of the present invention, the default resource module includes a unique number that represents and identifies the resource content contained within the default resource module. This unique number may be used to quickly verify whether the resource content of a default and current default resource module are the same and, therefore, compatible and safe to load. For example, in one embodiment the unique numerical identifier is a checksum of the resource contents within the module. According to one embodiment of the

MSFT\16734AM1.DOC

present invention, the checksum is based only on the resource content of a resource module. During the generation of the checksum, code and version information are excluded in the generation process, because a resource module may be updated with changes to its code or version number without any change to the resource content. This may be the case, for example, when a specific interface, such as a particular menu or dialog box, remains unchanged from one version of the operating system to the next, even though some of the underlying code may have changed.

Summary of Principal References Cited

The Rettig reference is commonly assigned to the assignee of this application, and describes certain aspects of an operating system that provides multilingual support. In particular, Rettig describes an operating system that provides for automatic redirection to the appropriate language-specific resources based on the user-selected language. In referring to appropriate language-specific resources, Rettig is referring to a resource that is in the user-selected language, and not in some other language that would be inappropriate for that user. Rettig does not describe, teach, or suggest comparing the resource contents of the underlying default and current default resource modules and loading the requested resource from the alternate resource module when the resource content is the same.

Hasset discloses a method of distributing information to multiple client devices on a network in which the use of a particular type of algorithm is used to compute a checksum, namely the message digest algorithm known as MD5. Nothing in Hasset describes, teaches, or suggests anything remotely related to multiple language user interfaces or operating systems that support multiple languages.

Rejection of Claims 10, 11, 25, 28, and 37 under 35 U.S.C. § 112, first paragraph

Claims 10, 11, 25, 28, and 37 have been amended to more particularly point out and distinctly claim the subject matter which the applicant regards as the invention. In particular, the claims as currently amended clarify that the resource content comparison is between a first and second resource contents of the default and current default resource modules. As noted in the

specification in the paragraph ending at the top of page 12, "a resource module may be updated with changes to its code or version number **without any change to the resource content.**" Since both the default and current default resource modules contain resources written in the same language, usually English, it is quite possible that the resource contents will be the same. Hence, in those cases where the first and second resource contents of the default and current default resource modules have not changed, then a comparison of those resources will reveal that they are the same, whether comparing the resource contents directly or by using a checksum or some other type of numerical representation of the contents. When they are the same, then resources from any alternate resource module localized from the default resource module may be safely loaded.

Rejection of Claims 1, 3, 4, 6-11, 13-16, 19, 25-28, 30-33, 36, 37, 39-42, and 44
under 35 U.S.C. § 112, second paragraph.

Claims 1, 3, 4, 6-11, 13-16, 19, 25-28, 30-33, 36, 37, 39-42, and 44 have been amended to more particularly point out and distinctly claim the subject matter which the applicant regards as the invention. In particular, the claims as currently amended clarify that the resource content comparison is between a first and second resource contents of the default and current default resource modules. The Examiner has asserted that the limitations of "default resource module" and "current default resource module" are not clearly defined. The Applicant respectfully disagrees.

As indicated in the specification in the third paragraph on page 11, "a default resource module 76 generally includes default resource content code and a variety of resources including data structures, templates, definition procedures, management routines, icon maps, and so forth associated with a particular resource, such as a menu, window, or dialog box, and is typically in English." The word "current" is used throughout the specification to refer to the most current version of the operating system. For example, in the first paragraph of the summary beginning on page 4 of the specification "a method is provided for dynamically verifying whether the resource content of a resource module is compatible with **the most current version of the**

operating system."   The very next paragraph on page 4 refers to the "current default resource module" to distinguish it from the "default resource module" from which the alternate resource modules were localized. That paragraph reads in pertinent part, "[t]he resource loader obtains the resource content of the default resource module from which the alternate resource module was localized. The resource loader obtains the resource content of the **current default resource module**. The resource loader compares the resource content of the default resource module from which the alternate resource module was localized with the resource content of the **current default resource module**. If the resource content of the default resource module from which the alternate resource module was localized is the same as the resource content of the **current default resource module**, the resource loader loads the alternate resource module."   This passage makes clear, as do other passages throughout the specification, that the "current default resource module" is a type of default resource module that is associated with the current version of the operating system, as opposed to the "default resource module," which is associated with the version of the operating system that was in effect when the localization of the alternate resource modules occurred.

The Applicant respectfully notes that the Examiner's supposition that "the default resource module is the Chinese version" if their PC is "running a Chinese Window version" is incorrect (Page 4 of Office Action, paragraph 8.) The specification makes clear that the alternate language versions of resources, such as menus or dialog boxes written in Chinese, are contained in the alternate resource modules, not the default resource modules (whether they are current default resource modules used in the current version of the operating system, or the default resource modules from which the alternate resource modules were localized). In addition, the Applicant respectfully notes that the word "version" is used throughout the specification to refer to the version of the operating system that is currently running, e.g., version 3.0 vs. version 3.1, and not the alternate language that may be used in the user interface.

<u>Rejection of Claims 1-8, 10-44 under 35 U.S.C. § 102(e) as being anticipated by Rettig</u>

The Examiner has cited numerous passages from Rettig to support a rejection of Claims 1-8, 10-44 as anticipated by Rettig. The Applicant respectfully disagrees.

Claim 1 as currently amended recites as follows:

1.   A method for dynamically verifying resource compatibility with an operating system, the method comprising:

obtaining a request to load a resource from an alternate resource module, wherein the alternate resource module corresponds to a selected interface;

obtaining a first resource content of a default resource module from which the alternate resource module was localized;

obtaining a second resource content of a current default resource module;

comparing the first resource content and the second resource content; and

loading the requested resource from the alternate resource module if the first resource content is the same as the second resource content.

As noted above, the Rettig reference is commonly assigned to the assignee of this application, and describes certain aspects of an operating system that provides multilingual support. Among other passages, the Examiner cites the passage in Col. 3, lines 7-17, that describes a Windows operating system that supports language specific libraries, such as those used to generate text messages in the current language based on a variable that indicates the user's locale. The Examiner contends that the requirement that the application that generates such a text message "identify precisely the appropriate language resource and where it is located" is equivalent to the limitations recited in Claim 1 of obtaining and comparing the resource contents of the default and current default resource modules, comparing them, and loading the requested resource from the alternate resource module if they compare the same. The Applicant respectfully disagrees.

"Identifying an appropriate language resource and where it is located" as used in Rettig refers merely to identifying one language from another, e.g., identifying a Chinese language resource from a French language resource, by using a variable that indicates the user's locale. "Identifying an appropriate language resource and where it is located" is not the same as obtaining and comparing the resource contents of default and current default resource modules to dynamically verify resource compatibility with an operating system as recited in Claim 1. As

noted above, in referring to appropriate language-specific resources, Rettig is referring to a resource that is in the user-selected language, and not in some other language that would be inappropriate for that user. Whether a resource is appropriate for a user because it is in the user-selected language is not the same as whether a resource is compatible with the user's operating system. Rettig does not describe, teach, or suggest comparing the resource contents of the underlying default and current default resource modules and loading the requested resource from the alternate resource module when the resource content is the same, i.e., when the resource is compatible with the operating system.

The remaining independent Claims 25, 28, and 37 recite similar limitations as recited in Claim 1, and are clearly and patentably distinguishable over Rettig for the same reasons set forth in the discussion of Claim 1. Claims 2-24, 26-27, 29-36, and 38-44 are allowable because they depend from allowable independent Claims 1, 25, 28, and 37, and because of their additional limitations. Consequently, reconsideration and allowance of Claims 1-8, 10-44 is respectfully requested.

Rejection of Claim 9 under 35 U.S.C. § 103(a) over Rettig in view of Hassett

The Examiner asserts that Hassett is analogous prior art, and that it teaches the use of the MD5 message digest algorithm, and that it would have been obvious to combine the teaching of Rettig and Hassett to obtain a checksum value calculated using and MD5 message digest algorithm as recited in Claim 9. The Applicant respectfully disagrees.

As noted above Hasset discloses a method of distributing information to multiple client devices on a network in which the use of a particular type of algorithm is used to compute a checksum, namely the message digest algorithm known as MD5. Nothing in Hasset describes, teaches, or suggests anything remotely related to multiple language user interfaces or operating systems that support multiple languages. Thus, Hassett is not analogous prior art.

Even if Hassett were analogous prior art, the MD5 message digest algorithm is used in Hassett to uniquely identify a data item, and not to compare resource contents as recited in Claim 9. Thus, the Examiner's reasoning that it would have been obvious to supplement Rettig with the teaching of Hassett "for the purpose of uniquely identifying any software" as set forth in
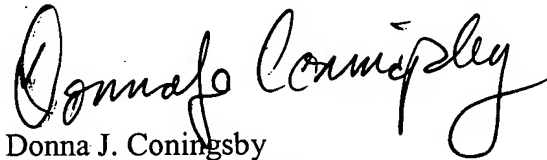
MSFT\16734AM1.DOC

page 20 of the Office Action fails to provide any motivation to combine the teachings of the references as required to establish a *prima facie* case. Nothing in Hassett suggests using the message digest algorithm for comparison purposes. Accordingly, the Examiner has failed to state a *prima facie* case of obviousness, and the rejection of Claim 9 should be withdrawn. Claim 9 is also allowable because it depends from allowable independent Claim 1. Consequently, reconsideration and allowance of Claim 9 is respectfully requested.

CONCLUSION

In view of the foregoing remarks, the Applicant submits that all of the claims in the present application are clearly patentably distinguishable over the teachings of Rettig and Hassett taken alone or in combination. Thus, Applicant submits that this application is in condition for allowance. Reconsideration and reexamination of the application, allowance of the claims, and passing of the application to issue at an early date are solicited. If the Examiner has any remaining questions concerning this application, the Examiner is invited to contact the Applicant's undersigned attorney at the number below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}

Donna J. Coningsby
Registration No. 41,684
Direct Dial No. 206.695.1719

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: _March 7, 2005_

DJC:pag

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue ·
Suite 2800
Seattle, Washington 98101
206.682.8100

-20-

MSFT\16734AM1.DOC